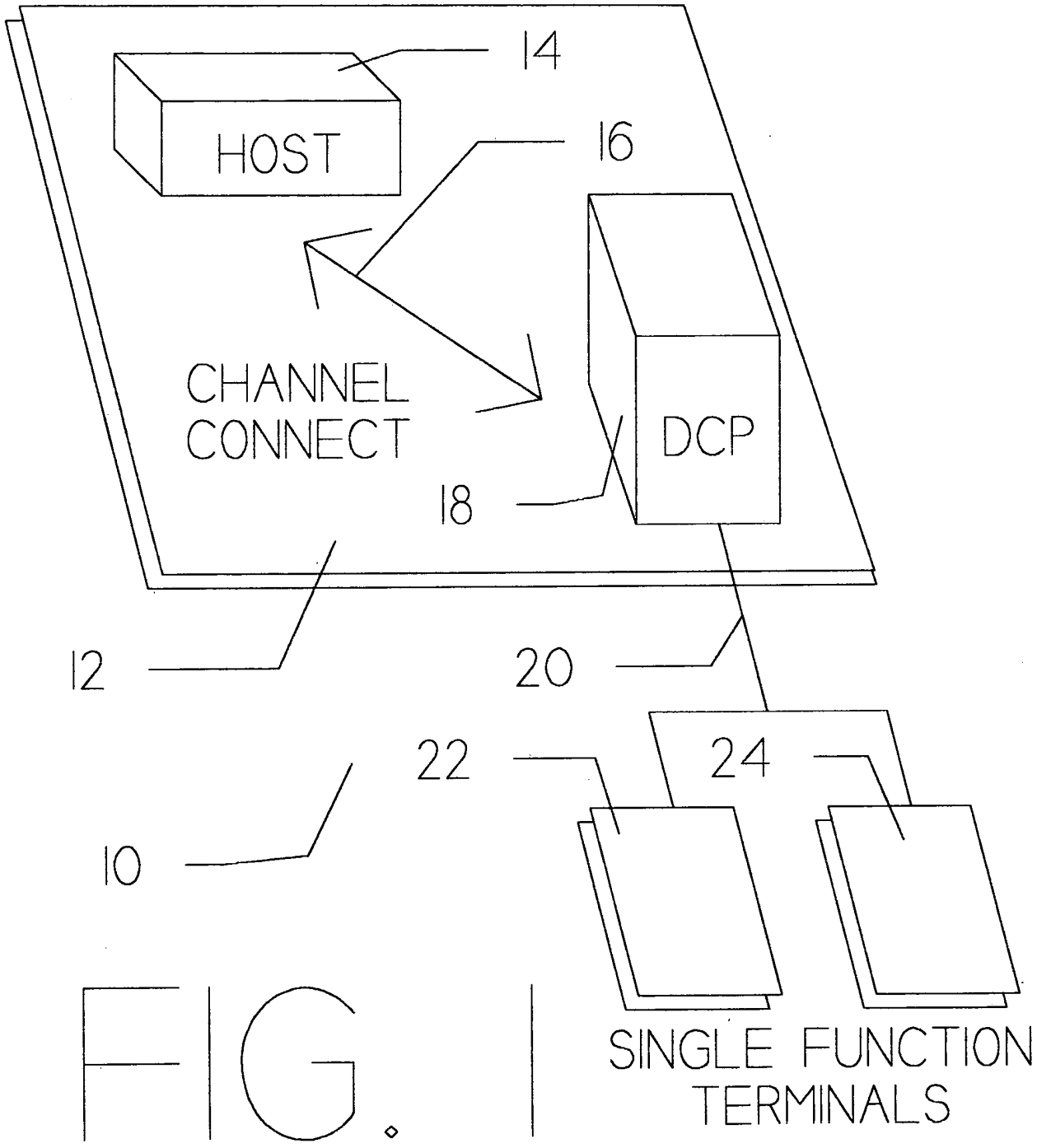
14
HOST
16
CHANNEL
CONNECT
DCP
18
12
20
22
24
10
SINGLE FUNCTION
TERMINALS

FIG. 1

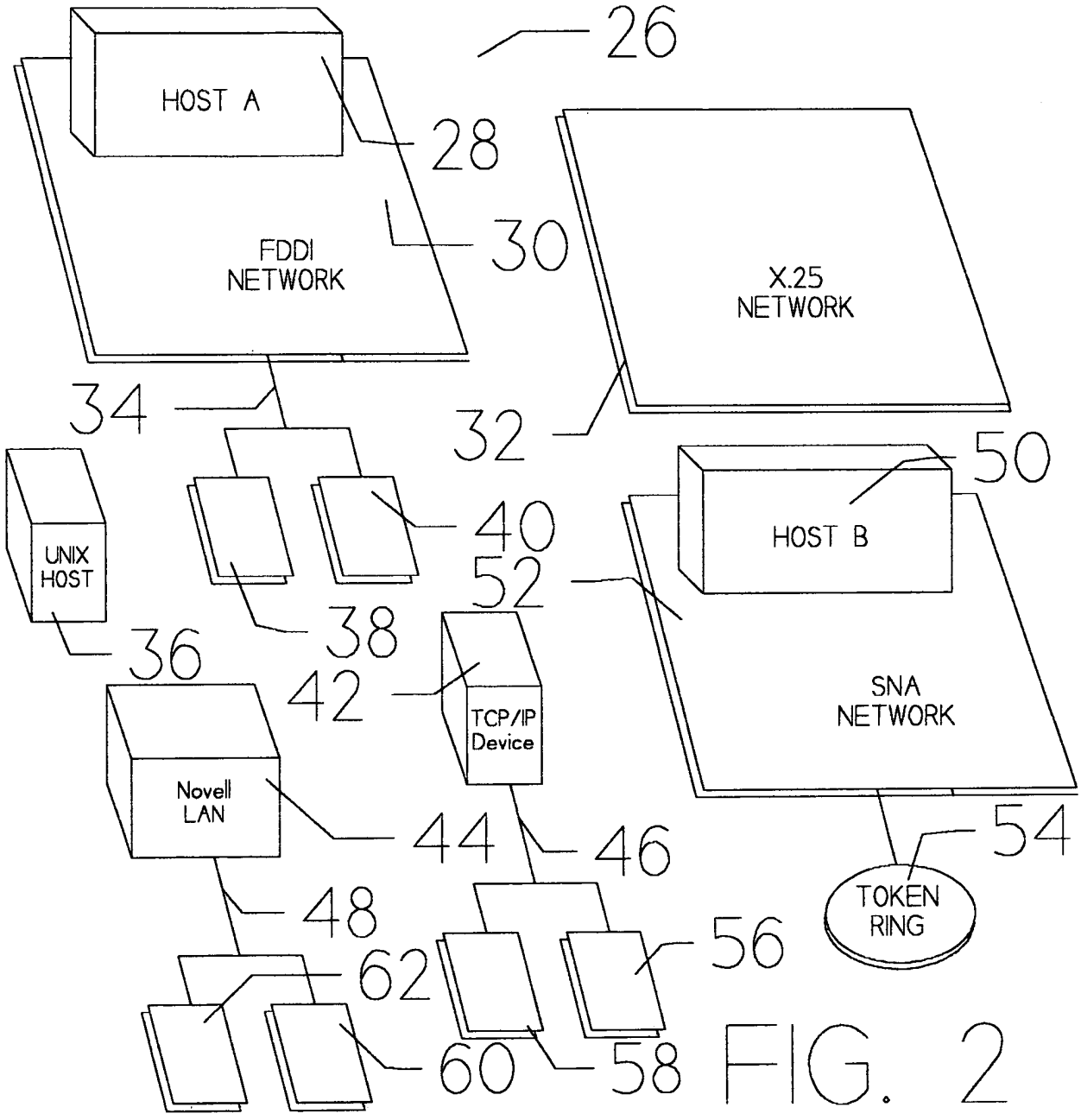
26
HOST A
28
30
FDDI
NETWORK
X.25
NETWORK
34
32
50
HOST B
40
UNIX
HOST
52
38
36
42
TCP/IP
Device
SNA
NETWORK
Novell
LAN
44
46
54
TOKEN
RING
48
56
62
60
58

FIG. 2

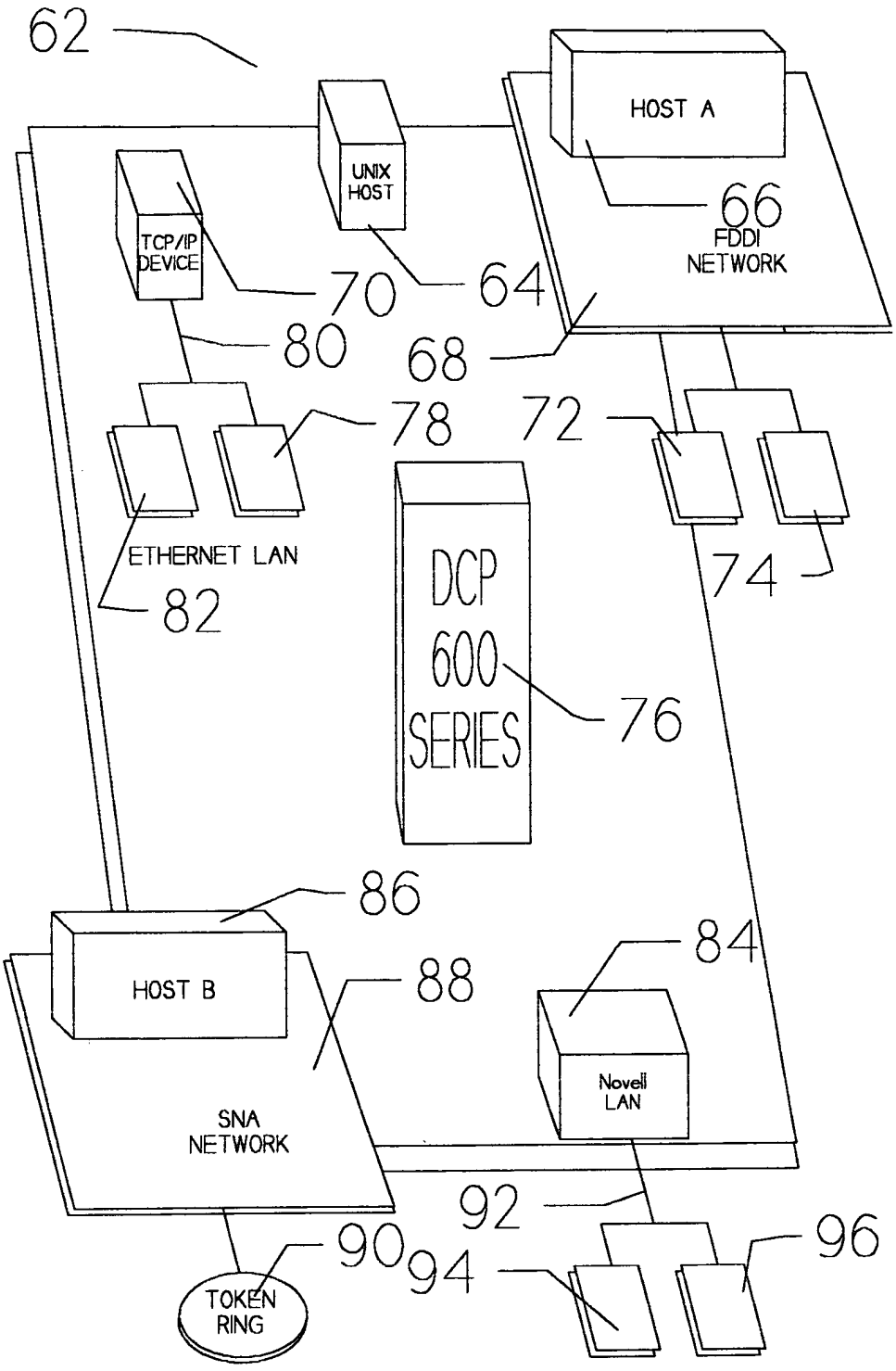
62
HOST A
UNIX HOST
TCP/IP DEVICE
66
FDDI NETWORK
70
64
80
68
78
72
ETHERNET LAN
82
DCP 600 SERIES
76
74
86
84
HOST B
88
Novell LAN
SNA NETWORK
92
90
94
96
TOKEN RING

FIG. 3

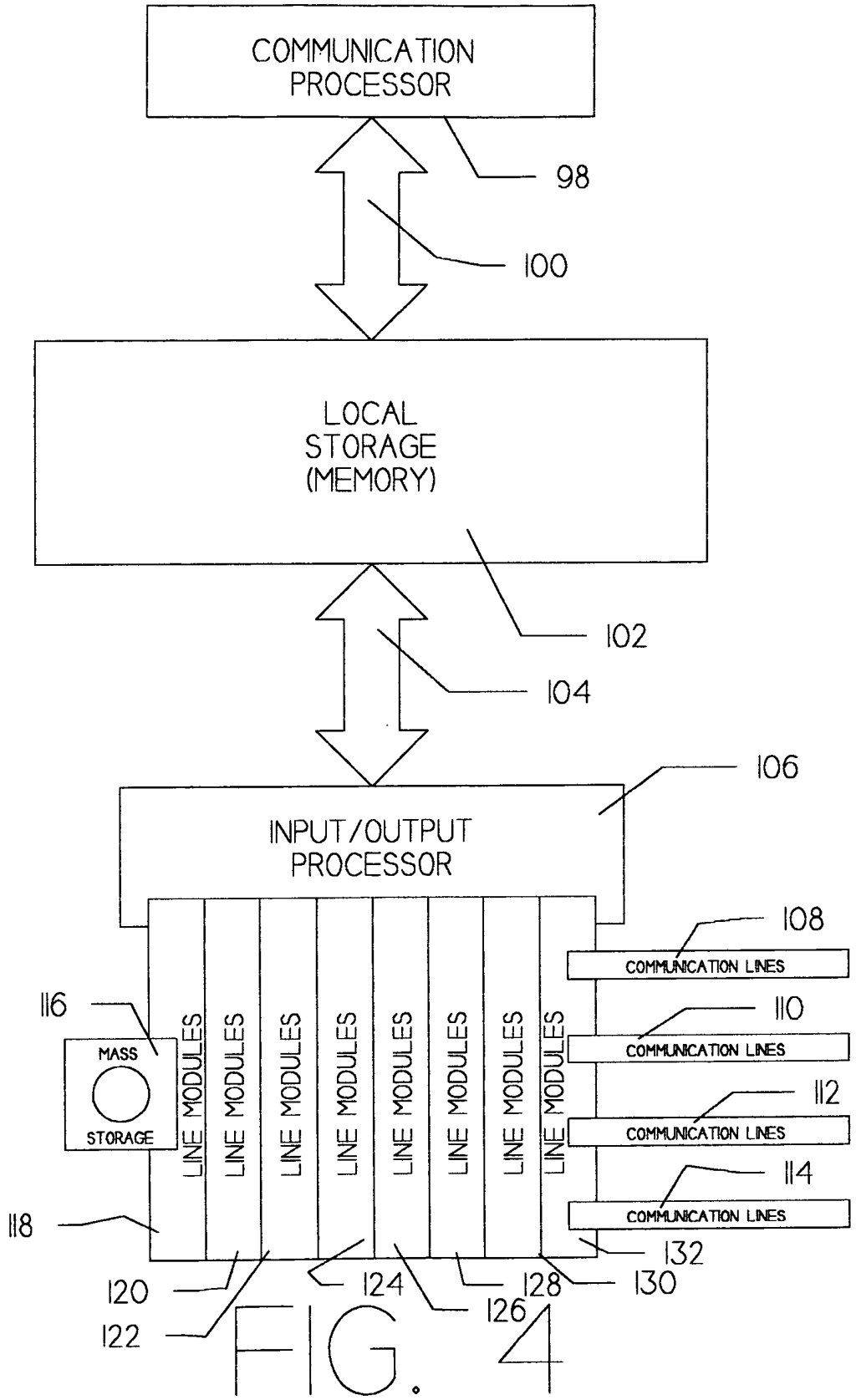
COMMUNICATION PROCESSOR
98
100
LOCAL STORAGE (MEMORY)
102
104
106
INPUT/OUTPUT PROCESSOR
108
COMMUNICATION LINES
110
COMMUNICATION LINES
112
COMMUNICATION LINES
114
COMMUNICATION LINES
116
MASS STORAGE
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
118
120
122
124
126
128
130
132

FIG. 4

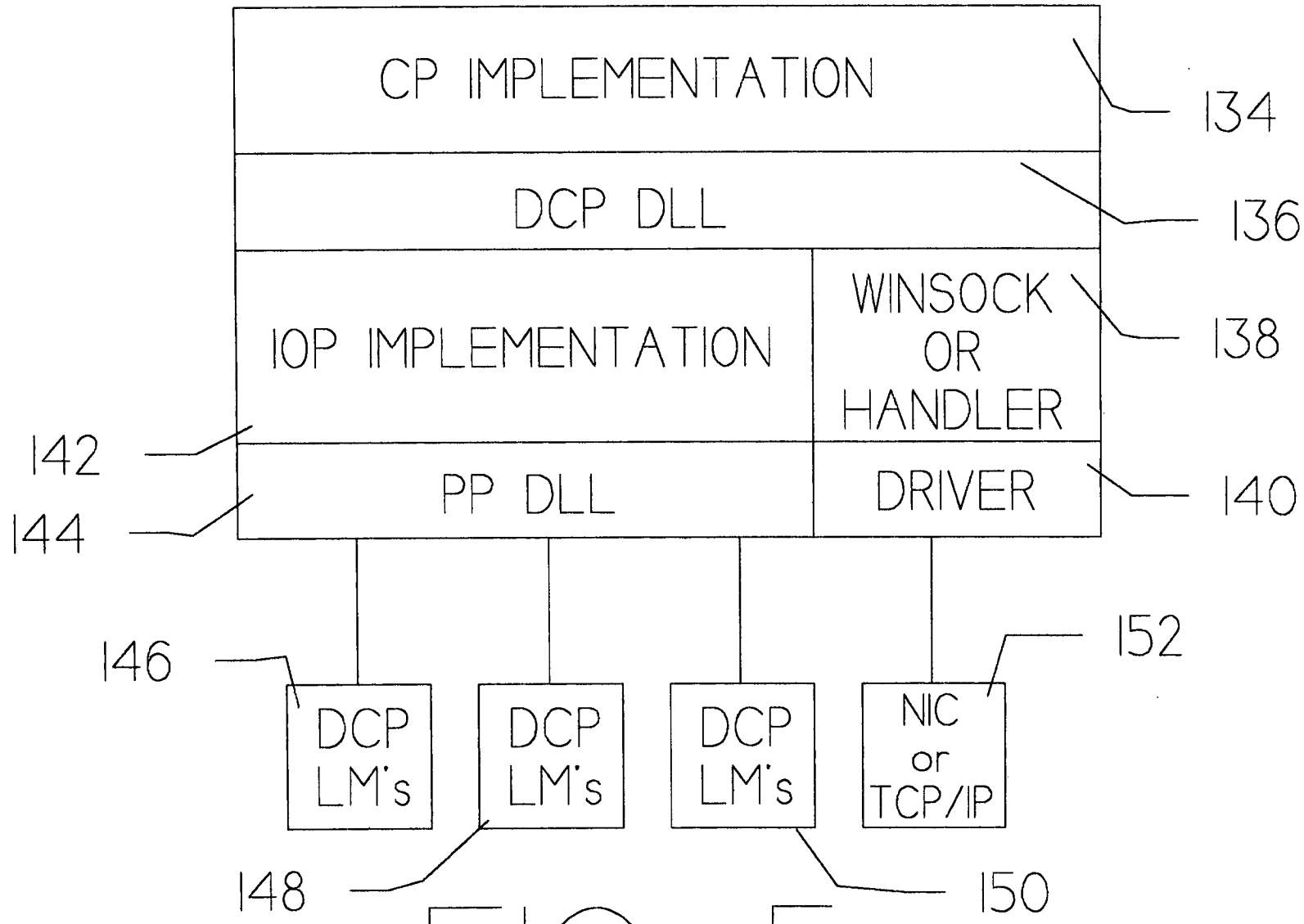

FIG. 5

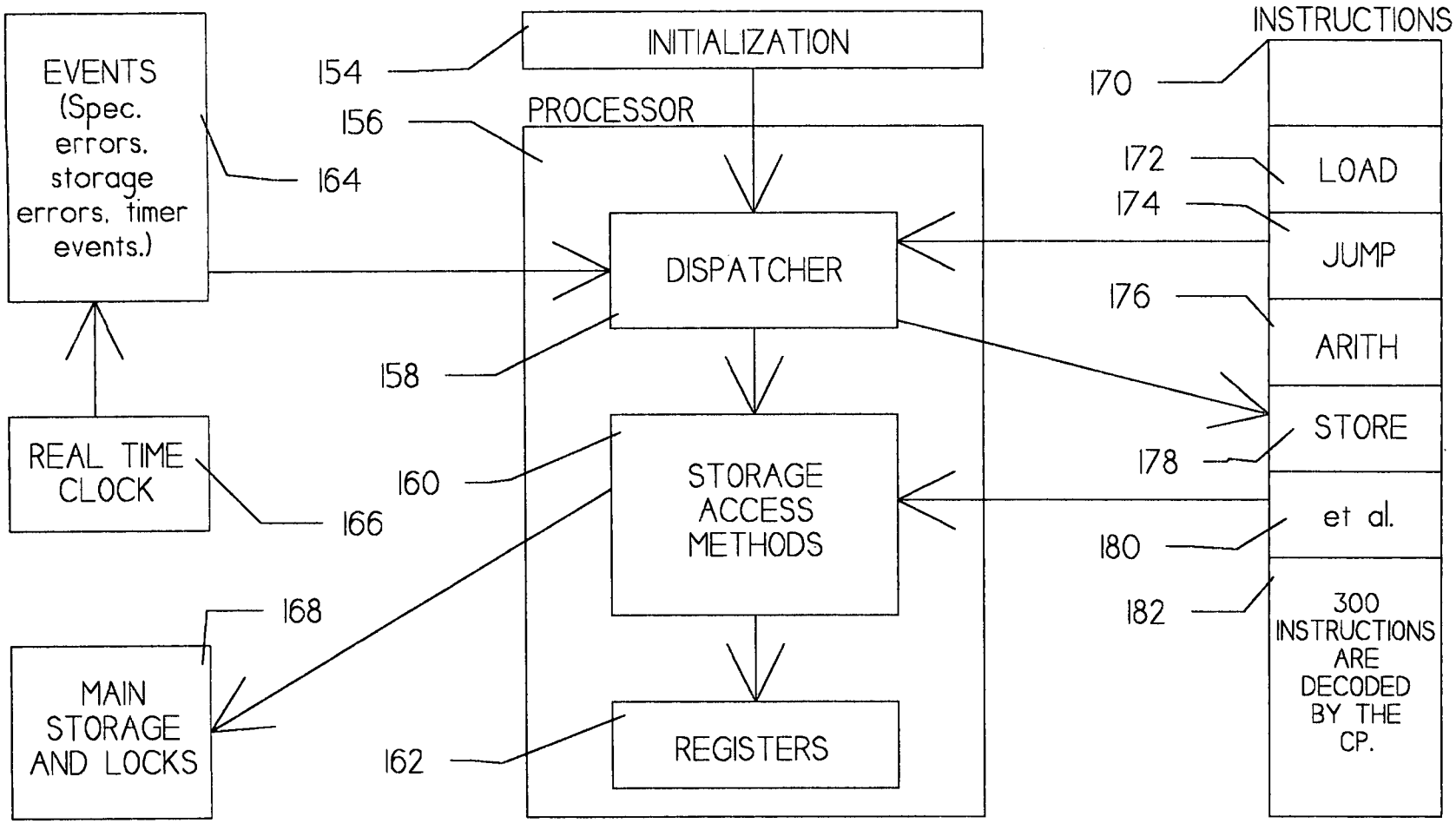

FIG. 6

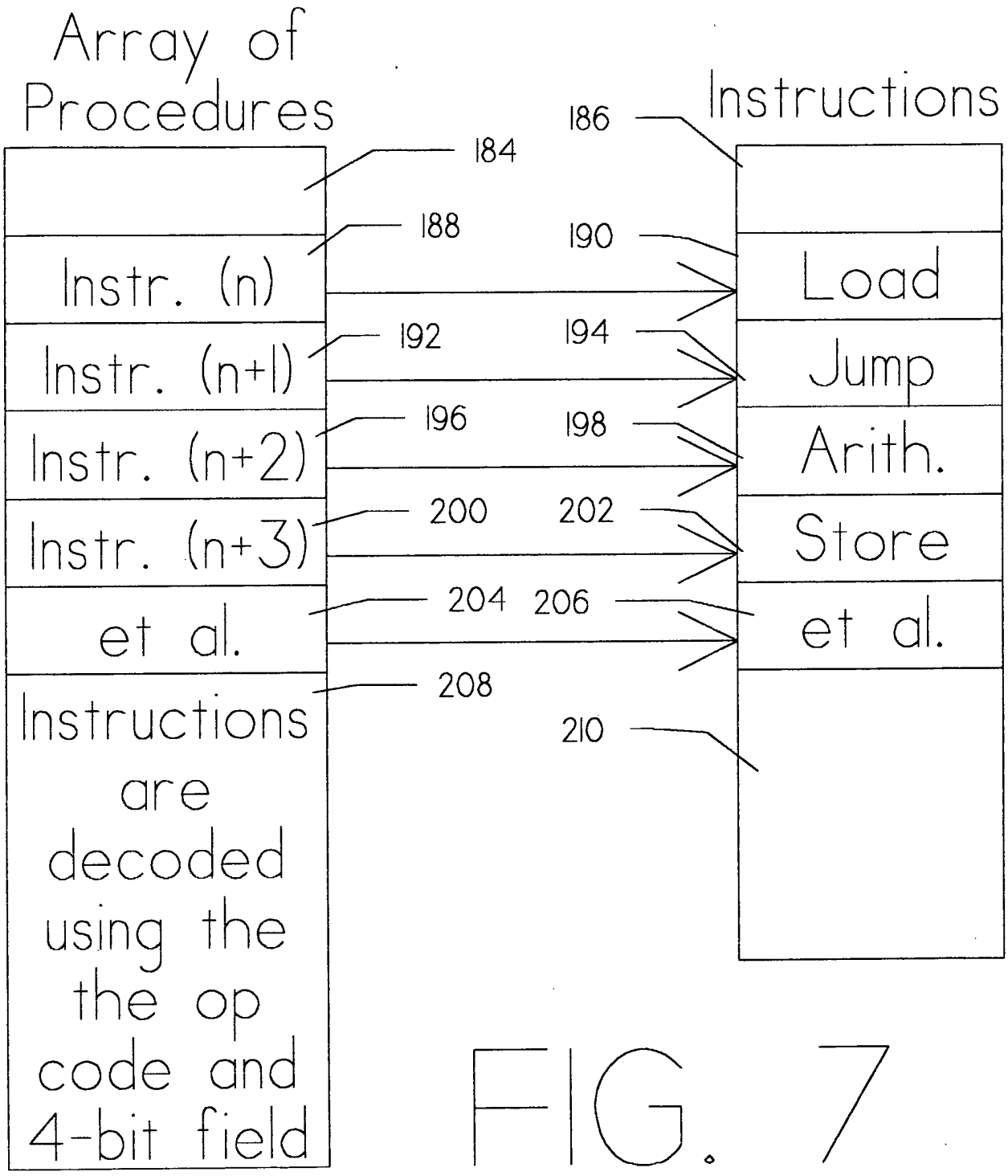
Array of Procedures
Instr. (n)
Instr. (n+1)
Instr. (n+2)
Instr. (n+3)
et al.
Instructions are decoded using the the op code and 4-bit field
184
188
192
196
200
204
206
208
210
Instructions
186
190
194
198
202
Load
Jump
Arith.
Store
et al.

FIG. 7

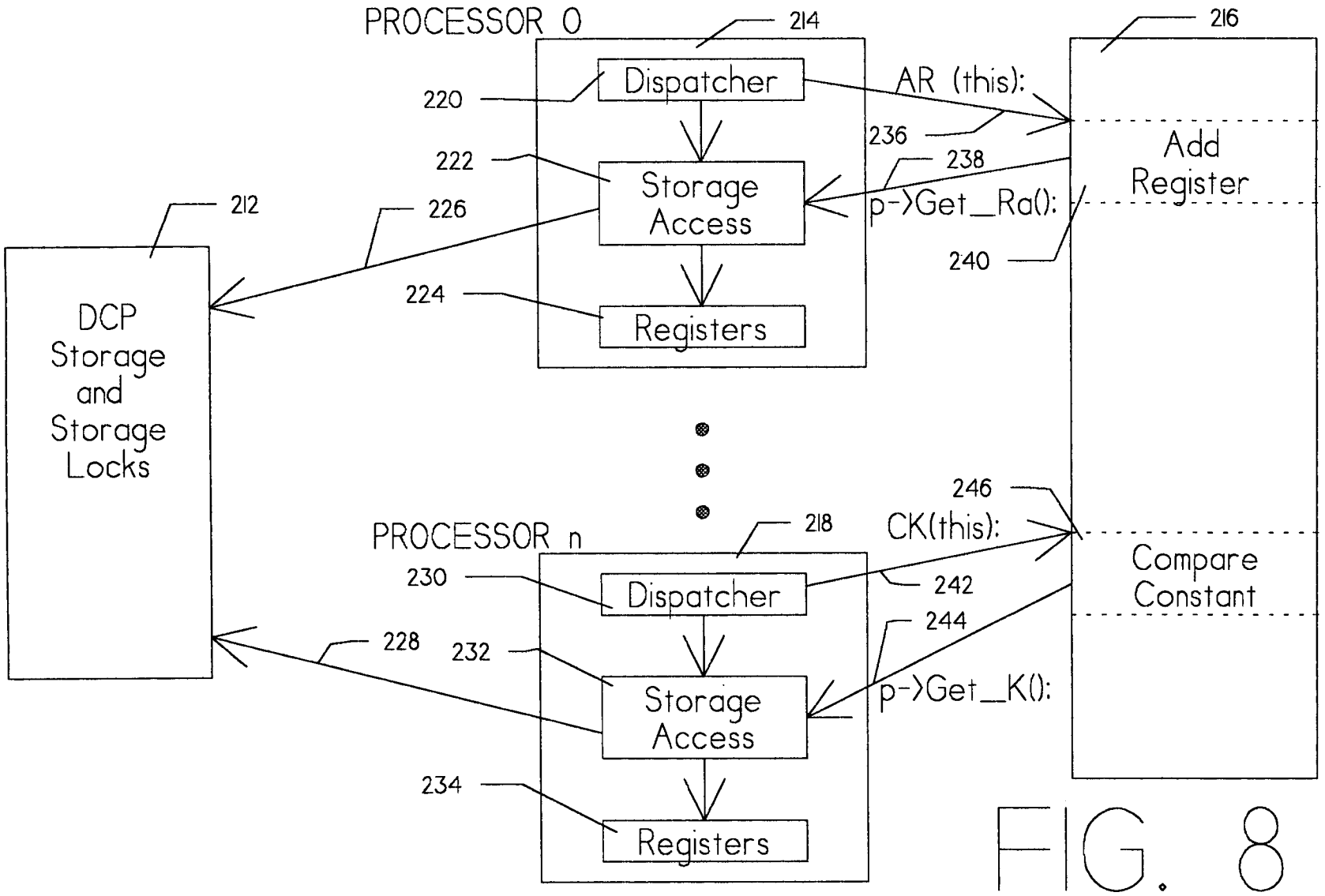
PROCESSOR 0
214
216
Dispatcher
220
AR (this):
236
222
238
Storage Access
Add Register
p->Get_Ra():
212
226
240
DCP Storage and Storage Locks
224
Registers
246
CK(this):
PROCESSOR n
218
Compare Constant
230
Dispatcher
242
244
228
232
Storage Access
p->Get_K():
234
Registers

FIG. 8

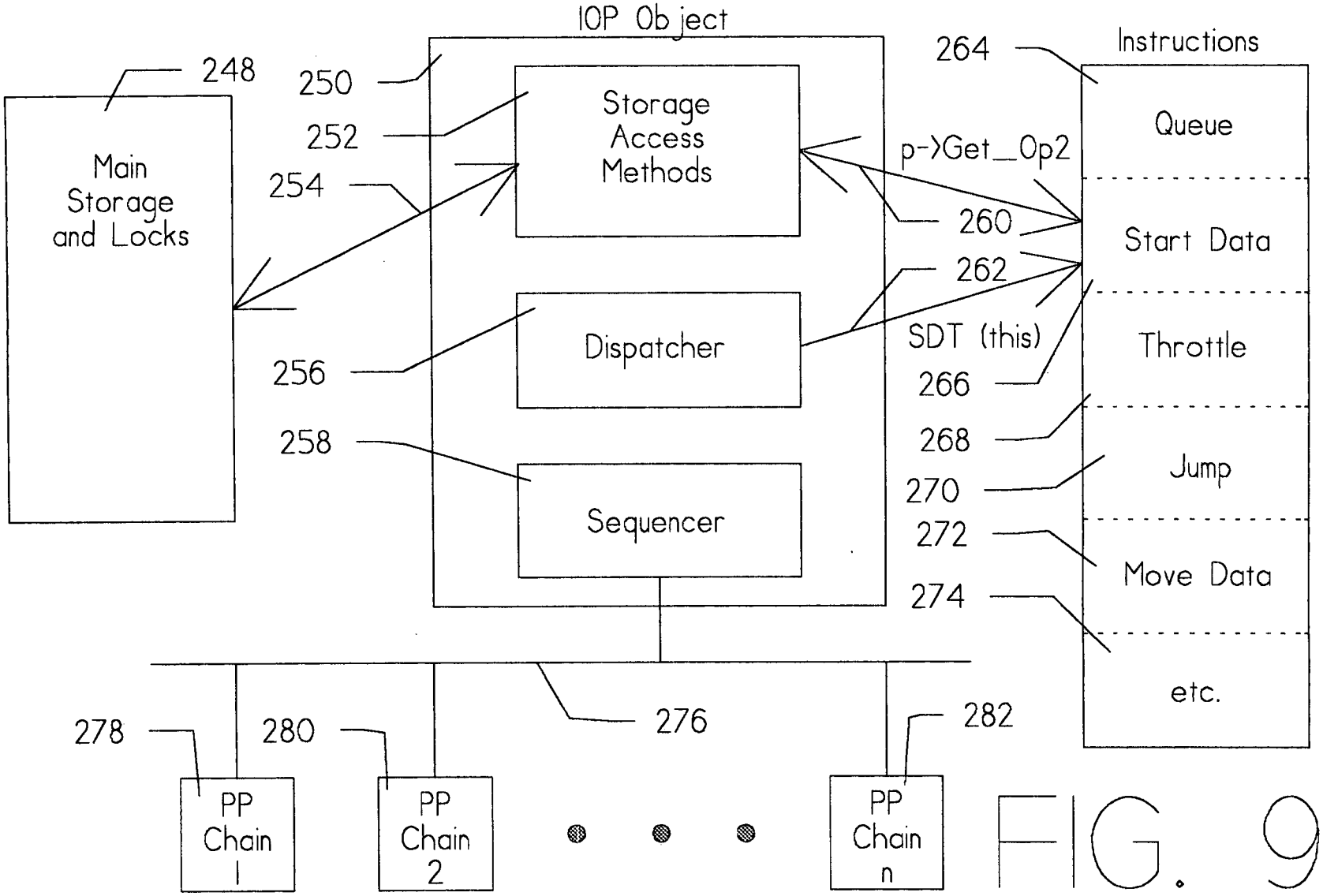

FIG. 9